

# IoT導入支援キットの作製

# IoT導入支援キット (Denshi-Pi V2.0) とは

## キットの構成と仕様

「見える化」の対象へ後付け可能

測定値や画像を遠隔で確認

本体: Raspberry Pi 4 Model B



「見える化」対象

各種  
センサ

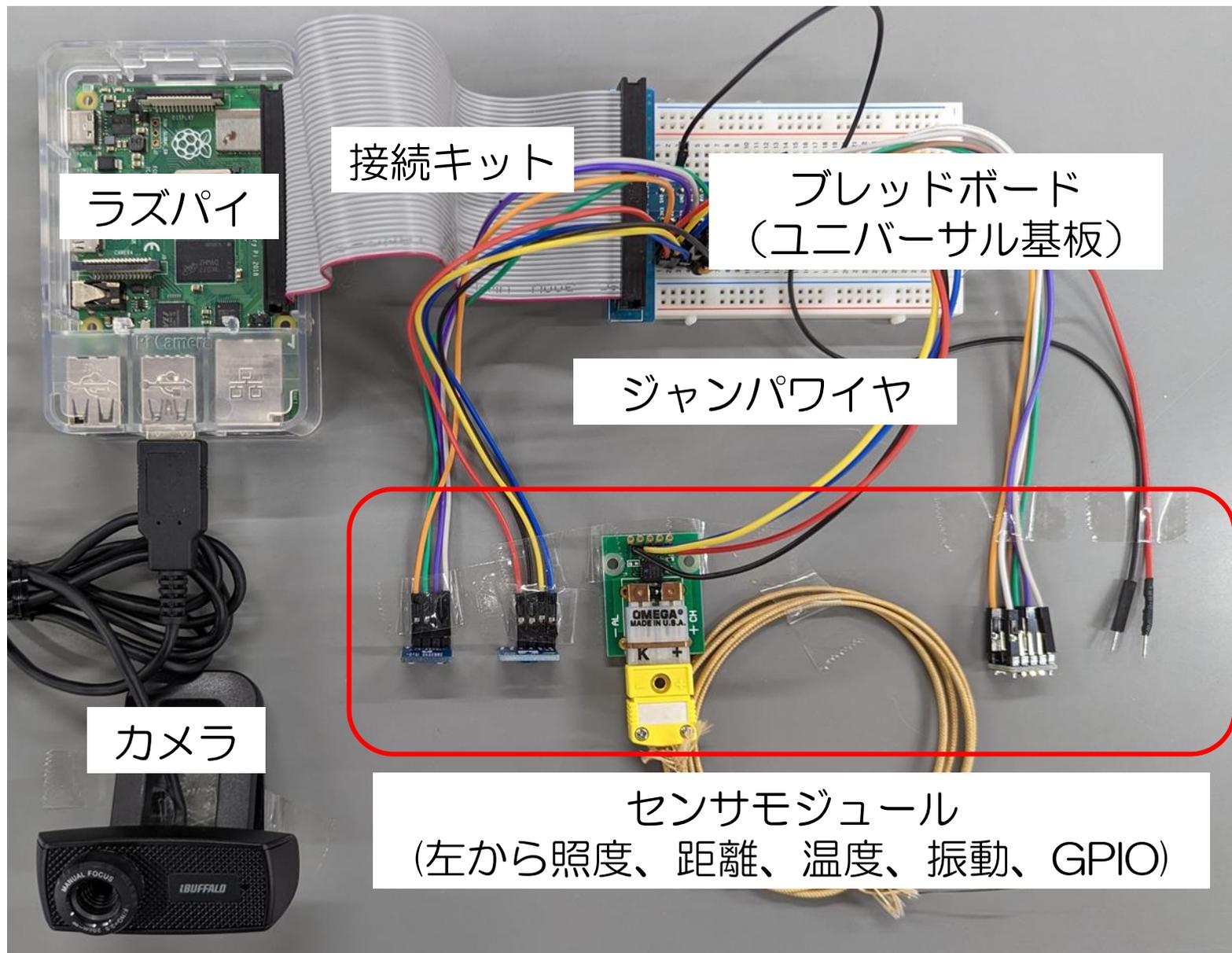


タブレット等

## キットの特徴

- PC、タブレット、スマートフォンで**遠隔操作・監視**  
各種センサ情報をPC、タブレット、スマホで確認。プログラム等の専門知識は不要。
- 既存設備に**後付けで設置可能**  
設備の改造等は基本的に不要。設備が稼働している状態でキットを設置可能。
- **安価**に構築可能(1キット部品費:約1.6万円)  
※2021年12月20日現在 半導体供給不足のため価格高騰中

# 全体概要 (ハードウェア)



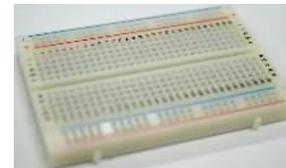
# 使用機材

1. ラズベリーパイ (Raspberry Pi 4 Model B)  
シングルボードコンピュータ
2. 温度センサ (MCP9600/ストロベリー・リナックス)  
熱電対温度センサモジュール
3. 振動センサ (MPU9250/ストロベリー・リナックス)  
9軸センサモジュール(3軸加速度センサを使用)
4. 距離センサ (VL53L0X/スイッチサイエンス)  
Time of Flight 距離センサモジュール
5. 照度センサ (TSL2561/ストロベリー・リナックス)  
照度センサモジュール



# 使用機材

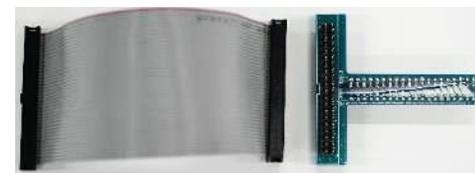
6. ブレッドボード (秋月電子通商)  
電子回路の試作、実験用基板



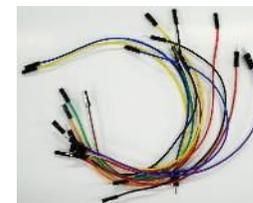
7. カメラ・マイク (BSWHD06MBK/Amazon)  
マイク内蔵120万画素USBwebカメラ



8. 接続キット (AE-RBPI-BOB40KIT/秋月電子通商)  
ラズパイとブレッドボードを接続



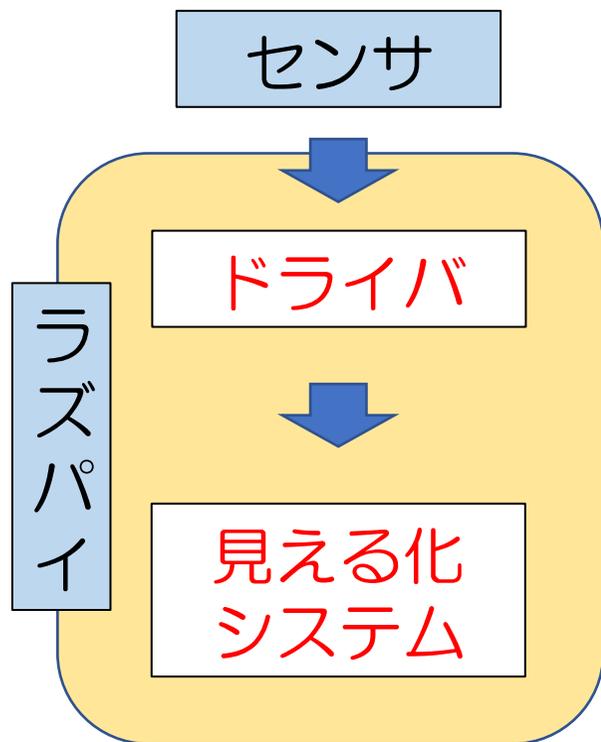
9. ジャンパワイヤ (18本) (Amazon)  
センサとラズパイを接続するケーブル



10. SDカード (Amazon)  
IoT導入支援キットの起動ディスク



# 全体概要（ソフトウェア）



## ドライバ (Python)

- センサからデータを取得
- Node-RED用にデータ形式を変換

## 見える化システム (Node-RED)

- ドライバからデータを取得
- データの見える化（ブラウザ表示）



## 遠隔操作（有線／無線LAN）

- ブラウザで遠隔操作・監視  
（PC、タブレット、スマートフォン）

# 本セクションのアジェンダ

- ① 起動ディスクの作製
- ② センサ回路の作製
- ③ 動作確認
- ④ 遠隔操作環境の構築

# ① 起動ディスクの作製

OS・ソフトウェアがインストールされた  
起動ディスク(SDカード)を作製

パソコン	ラズベリーパイ
ハードディスク	SDカード
↓	↓
OS ドライバー 各種ソフトウェア	OS ドライバー 各種ソフトウェア

# IoT導入支援キットの 起動ディスクの配布～起動ディスク作製について

## 起動ディスクの作製方法

福岡県工業技術センター  
ホームページ

圧縮された  
イメージ  
ファイル

ダウンロード



ご自身の  
Windowsのパソコン

- 解凍
- 起動ディスク作製  
(SDカードへの書込み)

SDカード  
挿入



ご自身の  
ラズベリーパイ  
システムの復元  
(使用領域拡張)

ダウンロードURL

<http://www.fitc.pref.fukuoka.jp/information/downloadpage.htm>

# IoT導入支援キットの 起動ディスクの配布～起動ディスク作製について

## 起動ディスクの作製方法

福岡県工業技術センター  
ホームページ

圧縮された  
イメージ  
ファイル

ダウンロード

## 次のスライド

ご自身の  
Windowsのパソコン

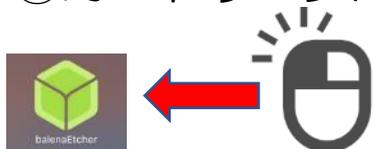
- 解凍
- 起動ディスク作製  
(SDカードへの書込み)

SDカード  
挿入

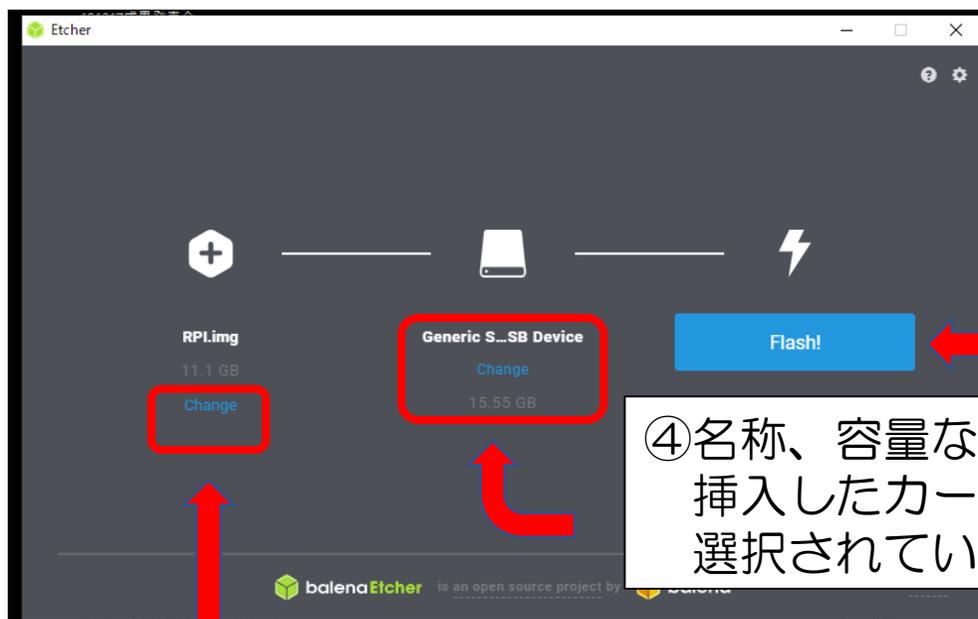
ご自身の  
ラズベリーパイ  
システムの復元  
(使用領域拡張)

# 起動ディスク作製

①カードリーダーにmicroSDカードを挿入し、PCにUSB接続する。



②アイコンをダブルクリックして、Etcherを起動



⑤Flash!をクリック

④名称、容量などから挿入したカードリーダーが選択されているか確認



③Select image をクリック  
→●●●.zip(ダウンロードしたファイル)を選択

# IoT導入支援キットの 起動ディスクの配布～起動ディスク作製について

## 起動ディスクの作製方法

次のスライド

福岡県工業技術センター  
ホームページ

圧縮された  
イメージ  
ファイル

ダウンロード

ご自身の  
Windowsのパソコン

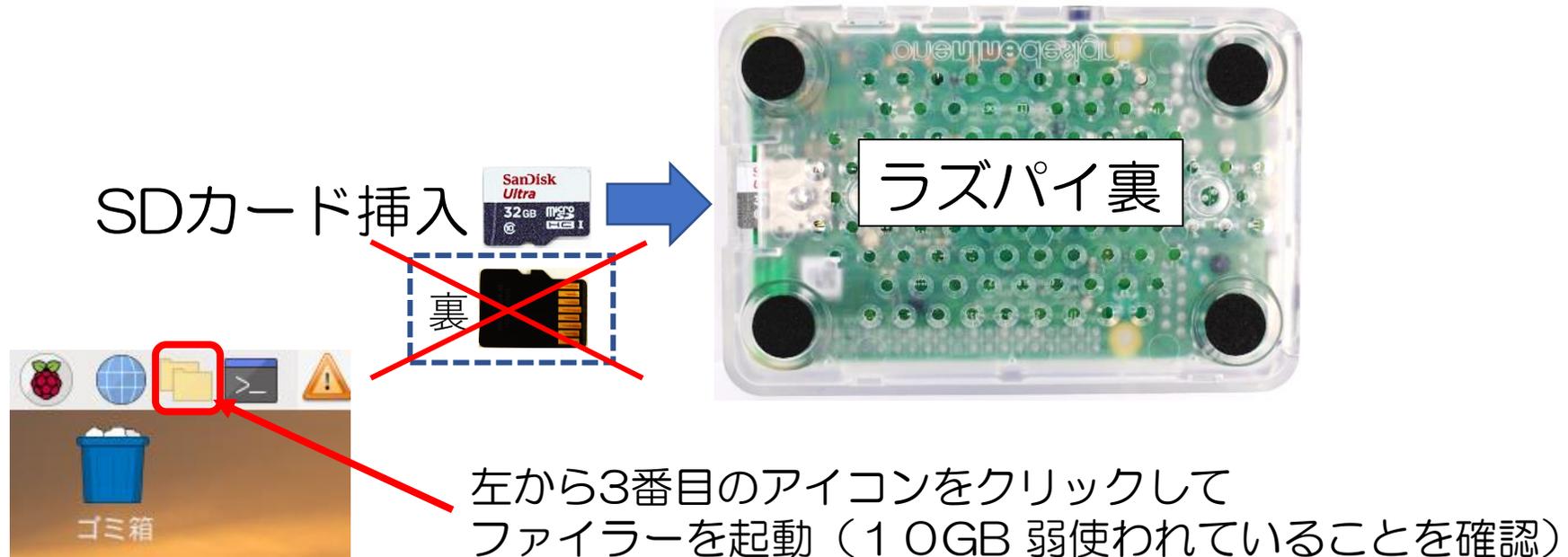
- 解凍
- 起動ディスク作製  
(SDカードへの書込み)

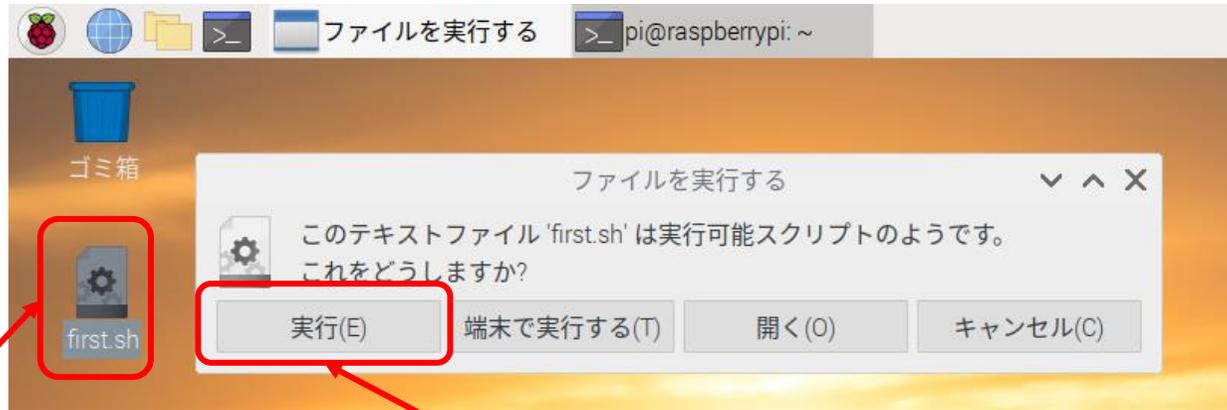
SDカード  
挿入

ご自身の  
ラズベリーパイ  
システムの復元  
(使用領域拡張)

# 使用領域拡張

- ①SDカードをラズパイに挿入
- ②HDMIケーブルを接続
- ③モニタ用ACアダプタを接続
- ④キーボード、マウスを接続
- ⑤ラズパイ用ACアダプタ（USB-C）を接続 ⇒ ラズパイ起動





①first.sh をダブルクリック    ②実行をクリック



自動で再起動されます。



左から3番目のアイコンをクリックして  
ファイラーを起動（30G弱使われていることを確認）

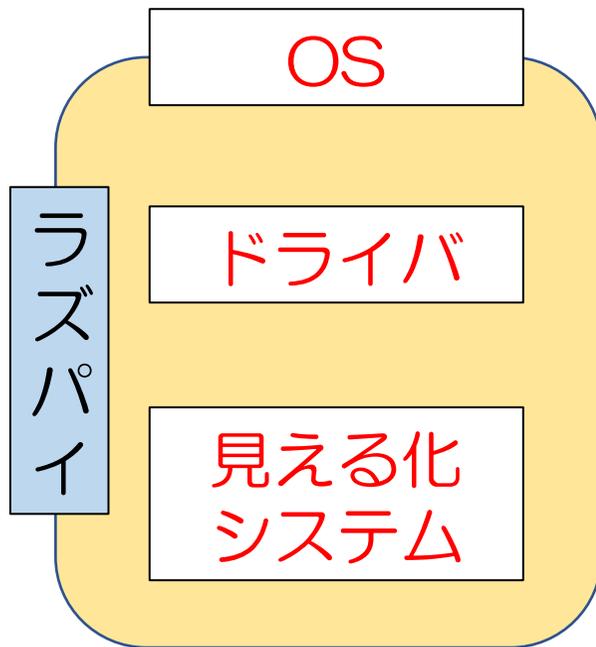
first.shはもう使わないので、ゴミ箱に捨ててください。



一番左のアイコンをクリックして  
一番下のshutdown→shutdownを選択

## 起動ディスクの作製完了

- OS(Raspberry Pi OS)
- ドライバ(Python)
- 見える化システム(Node-RED)



### ドライバ (Python)

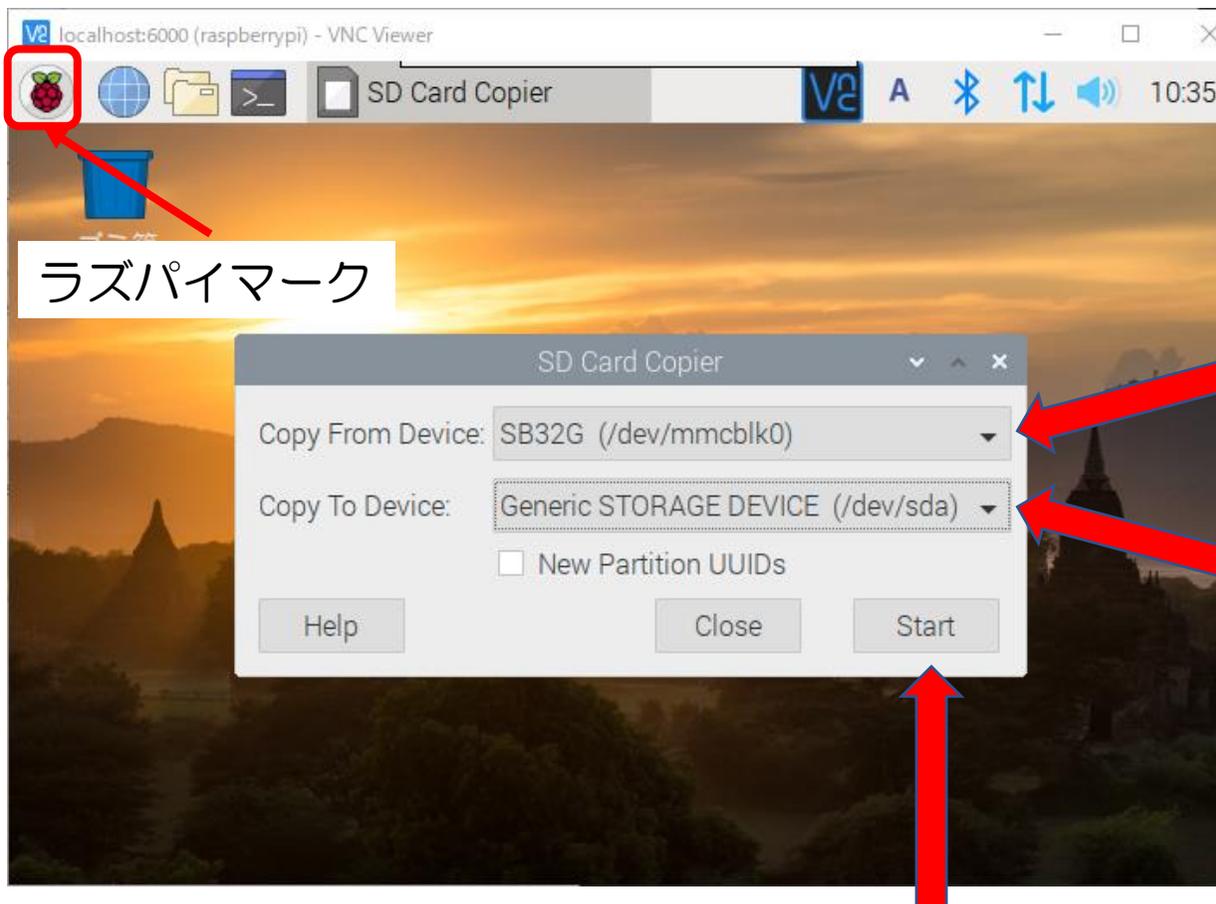
- センサからデータを取得
- Node-RED用にデータ形式を変換

### 見える化システム (Node-RED)

- ドライバからデータを取得
- データの見える化 (ブラウザ表示)

## (補足資料) SDカードからSDカードへのコピー

ラズパイにmicroSDカードを挿したUSBカードリーダーを接続し『SDCardCopier※』を実行。



- ①ラズパイマークをクリック。
- ②アクセサリにカーソルを合わせる。
- ③SDCardCopierをクリック。

④

SB32G(/dev/mmcbk0)  
を選ぶ。

⑤

Generic STORAGE DEVICE  
(/dev/sda)を選ぶ。

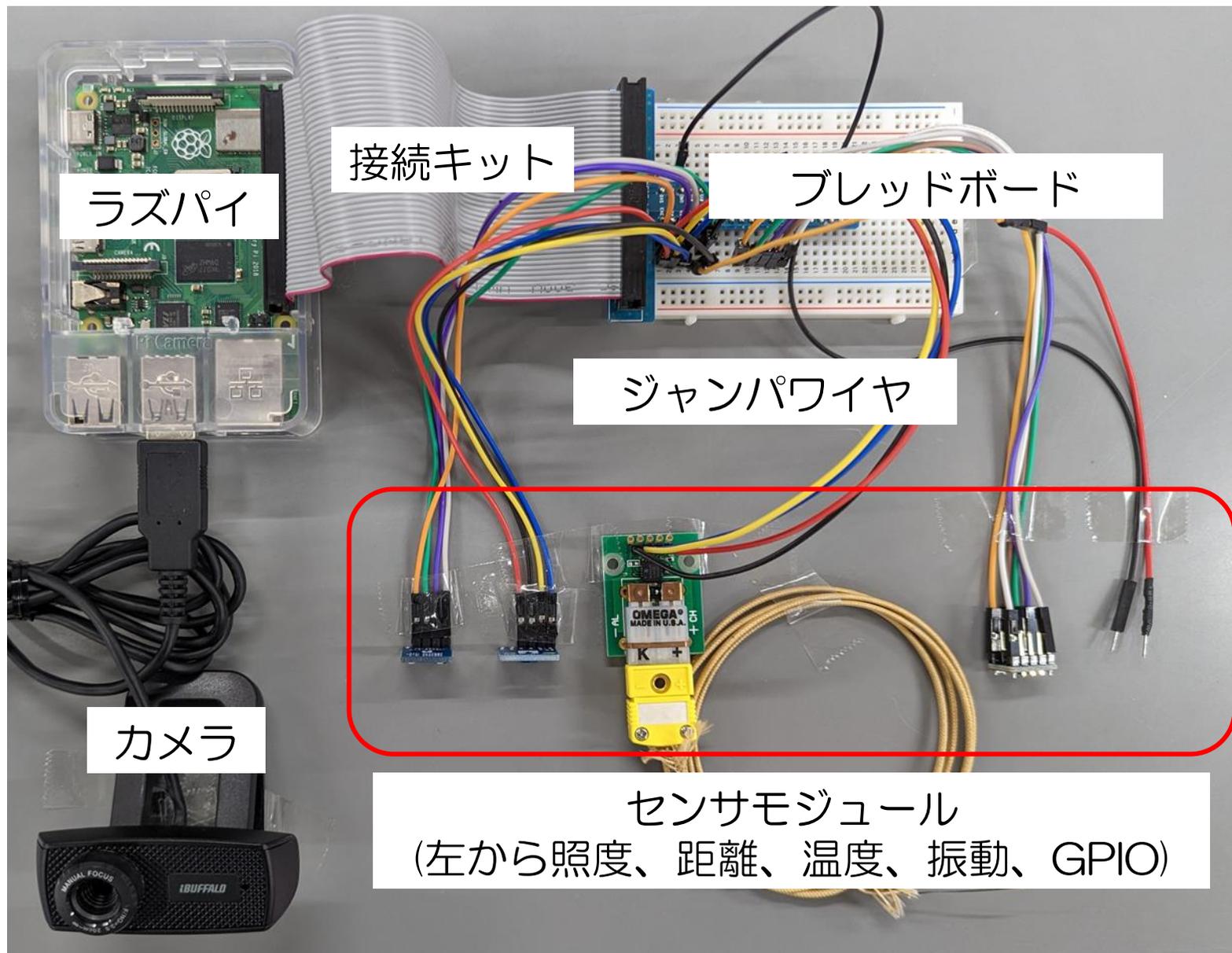
⑥ Startを押して実行する。

- ※1 終了まで15分程度かかります。
- ※2 複製に使うmicroSDカードはコピー元と同じか大きいものを選んで下さい。
- ※3 同じ容量表記のSDカード同士でも容量が異なりコピーできない場合があります。

## ② センサ回路の作製

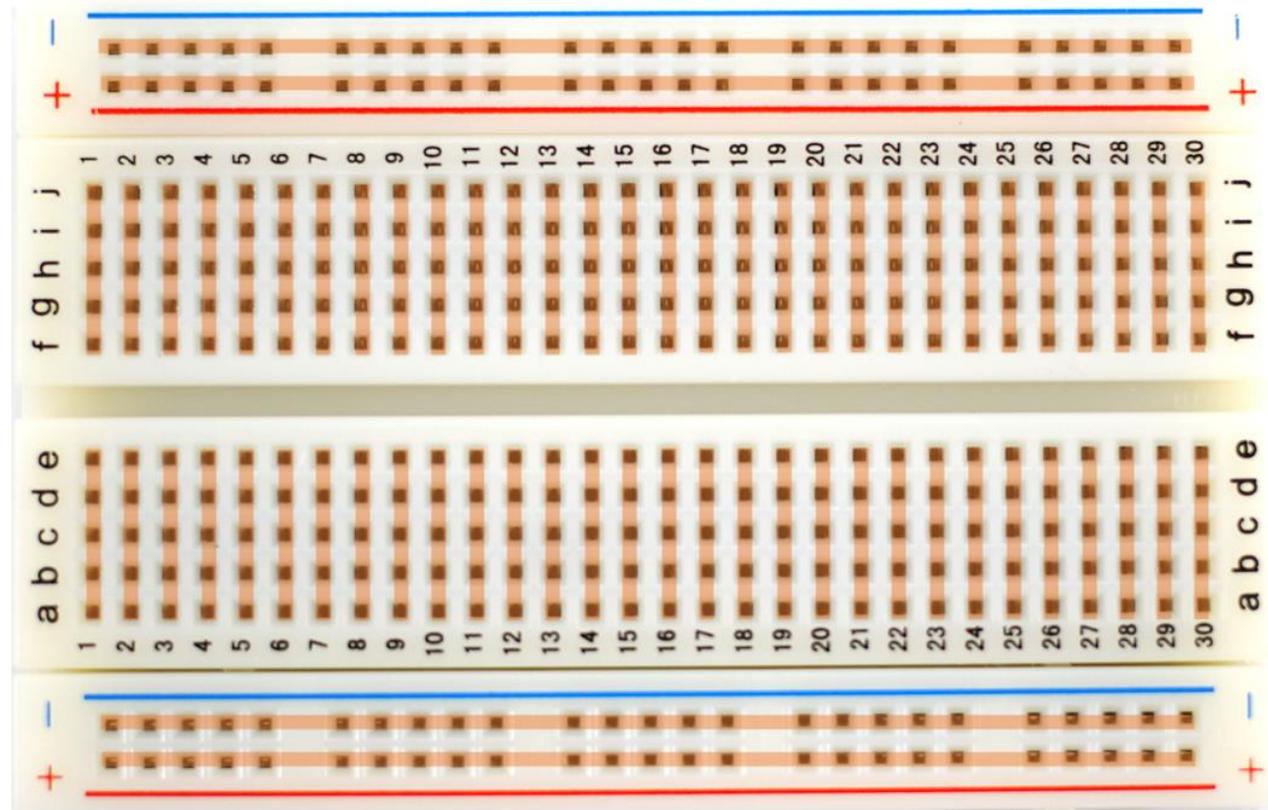
センサ接続用の回路を作製

# 全体概要 (ハードウェア)



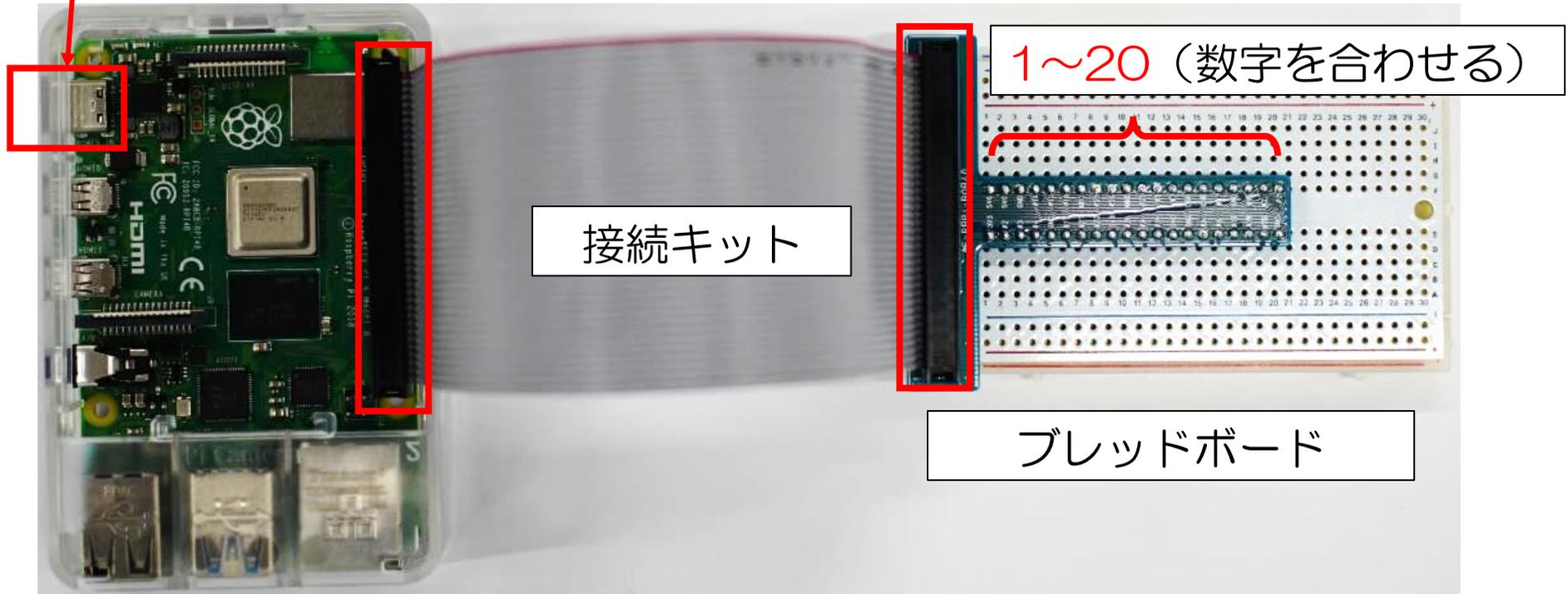
# ブレッドボードについて

- オレンジ色の部分が内部で導通
- ジャンプワイヤを直接ブレッドボードに接続（半田付け不要）



# センサ回路の作製

USB-Cのアダプターを抜いてから  
接続キットをブレッドボードとラズパイに接続

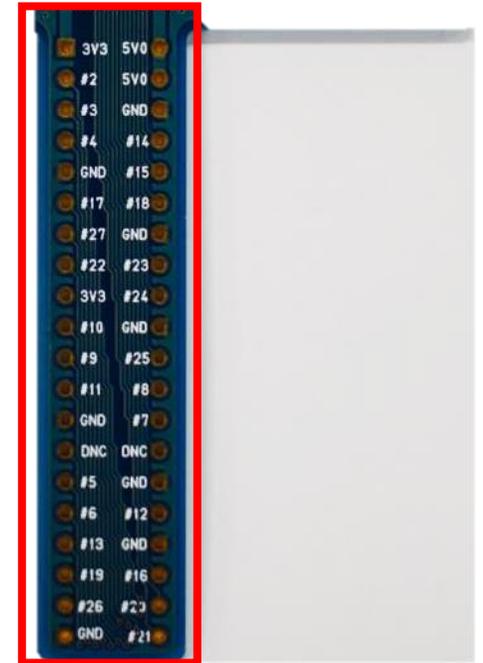


# ピンアサインについて(ラズパイ)

I2C通信で使用するポート (温度・距離・照度センサ)

3.3V Power - 1	2 - 5V Power
SDA1 - GPIO02 - 3	4 - 5V Power
SCL1 - GPIO03 - 5	6 - Ground
GPIO04 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO09 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21

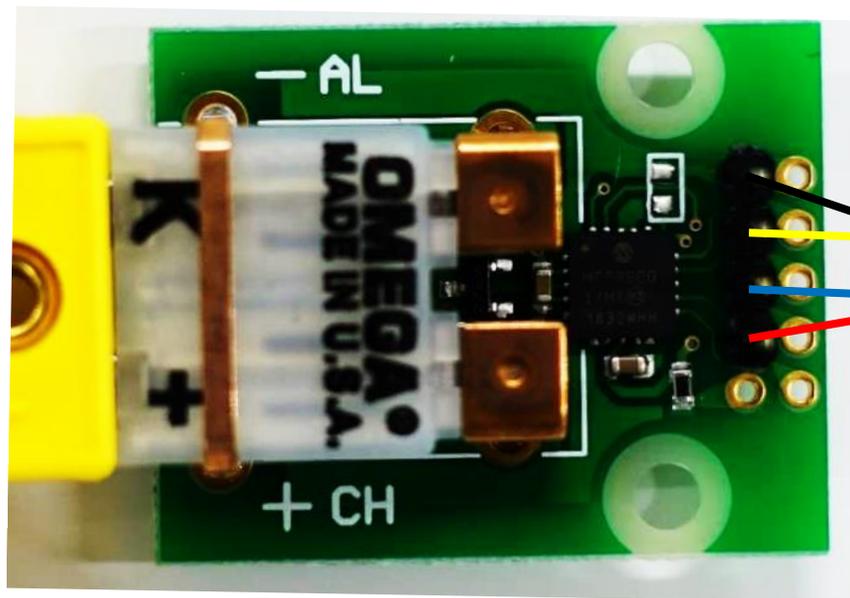
接続キット



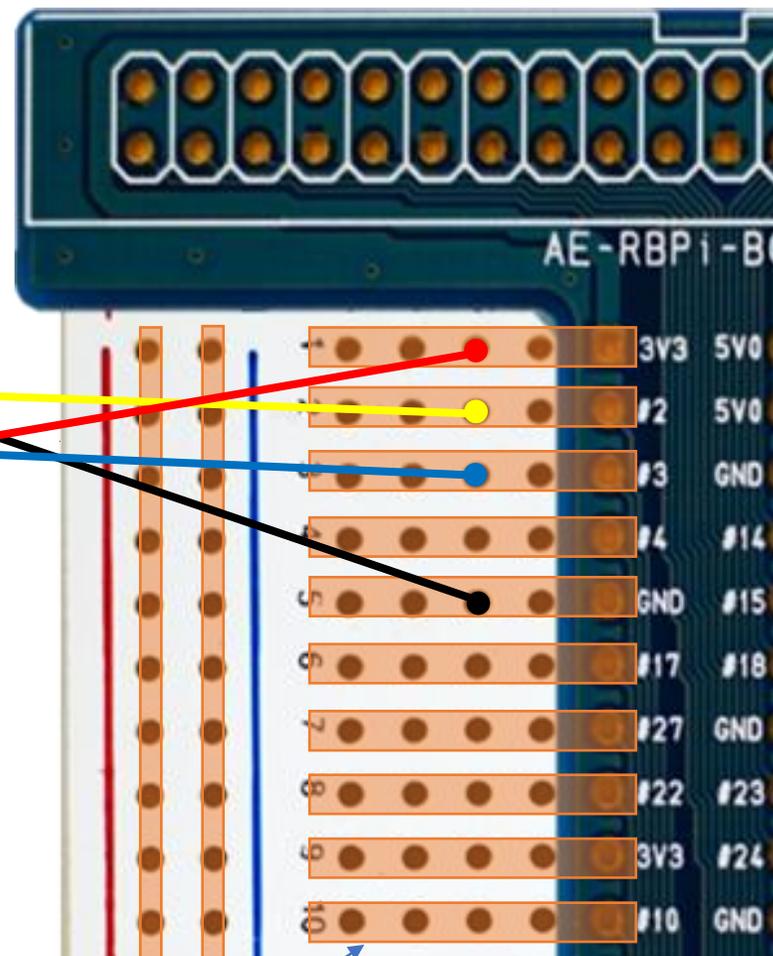
GPIO通信で使用するポート  
(無電圧接点)

SPI通信で使用するポート (振動センサ)

# 温度センサ接続(I2C通信)

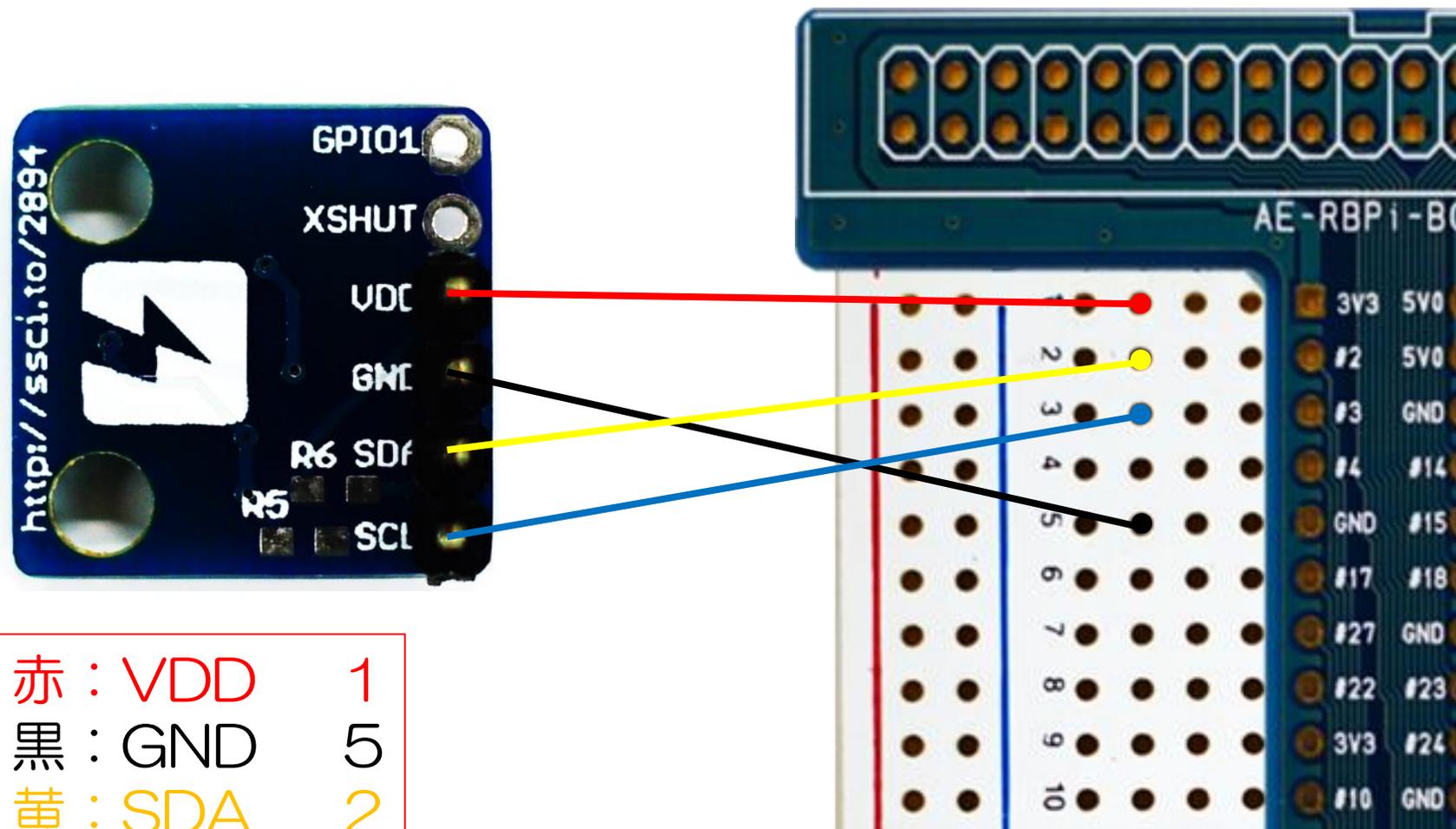


黒	: GND	5
黄	: SDA	2
青	: SCL	3
赤	: VDD	1



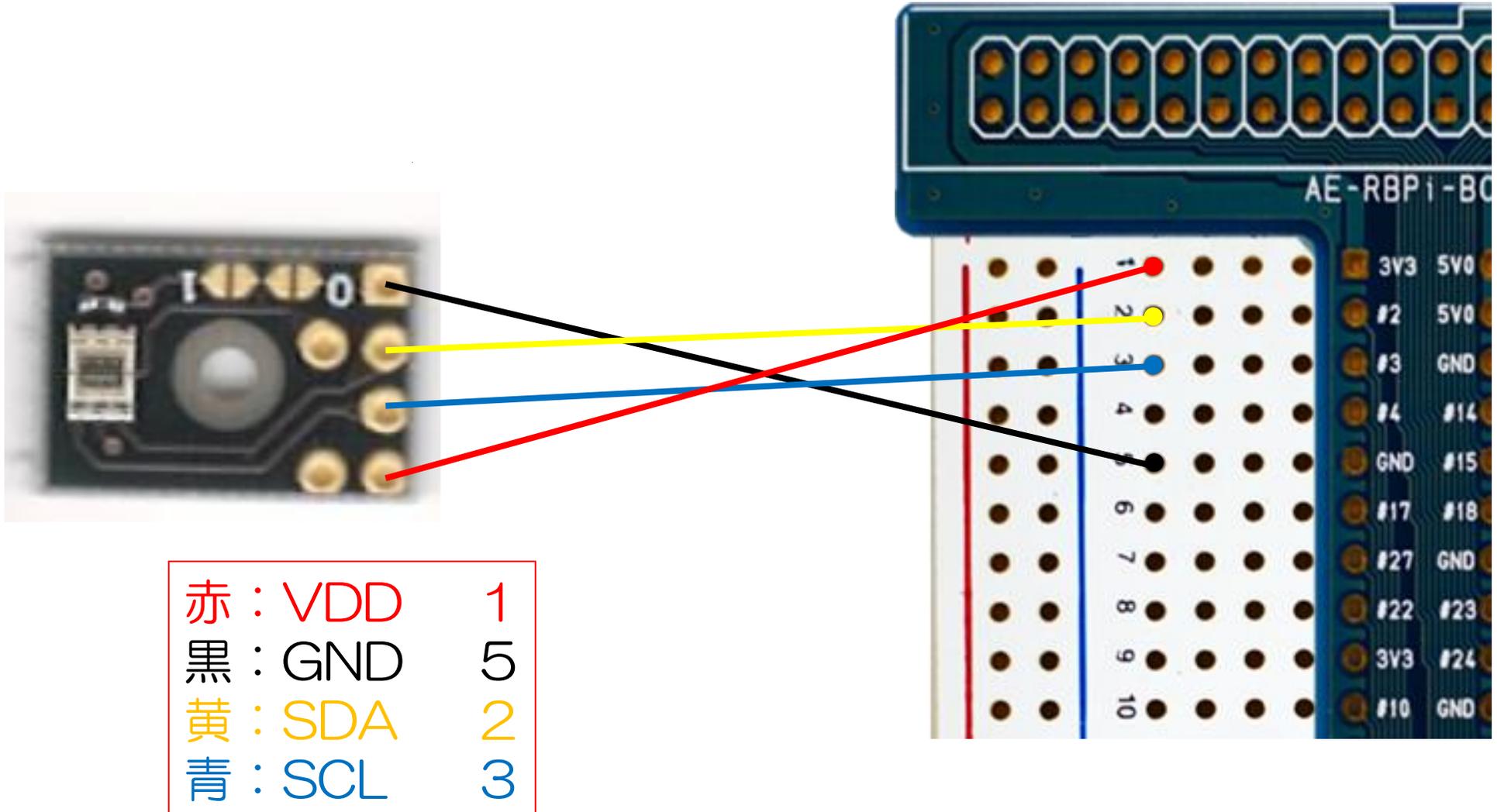
背面の導通パターン

# 距離センサ接続(I2C通信)

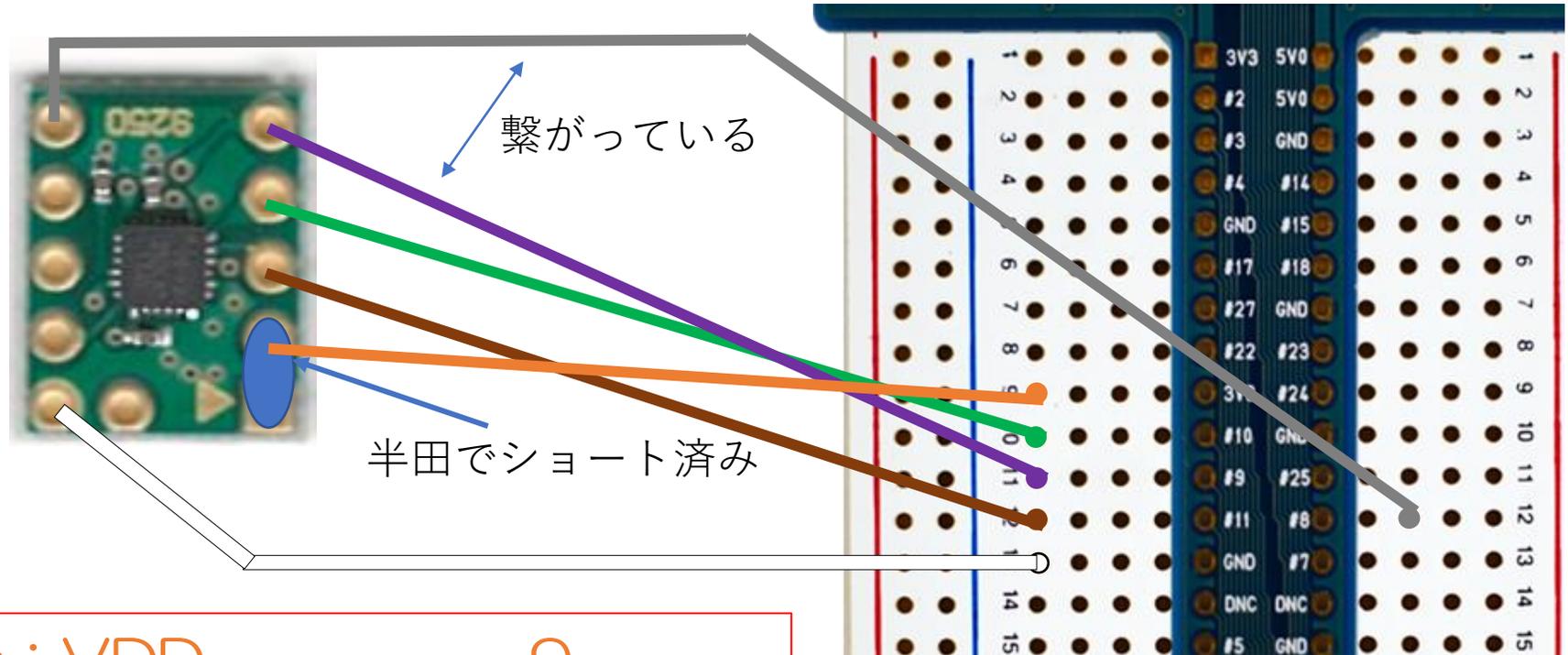


赤	: VDD	1
黒	: GND	5
黄	: SDA	2
青	: SCL	3

# 照度センサ接続(I2C通信)

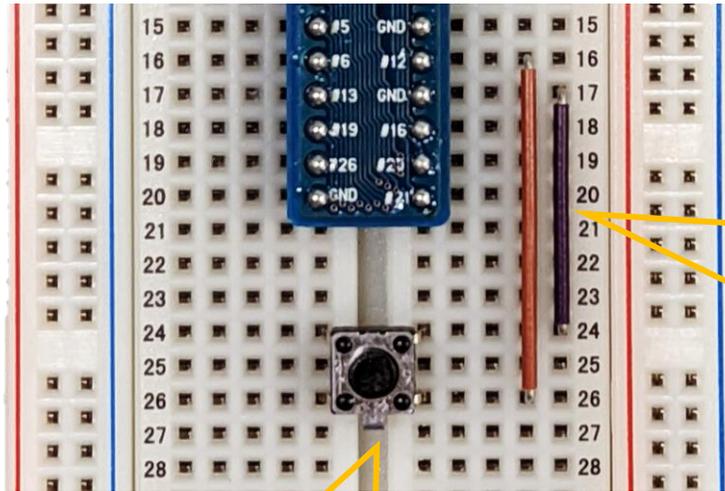


# 振動センサ接続 (SPI通信)

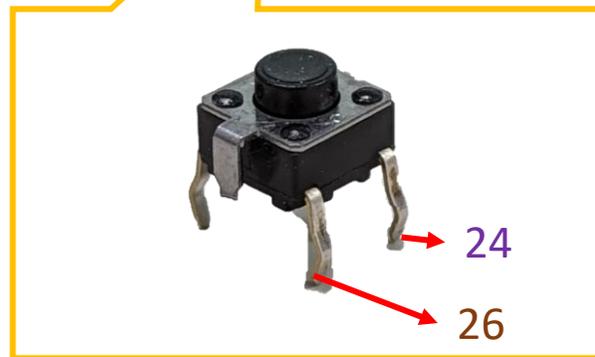


橙	: VDD	9
白	: GND	13
茶	: SCL	12
緑	: SDA	10
紫	: ADO/SDO	11
灰	: NCS	12(右側)

# 無電圧接点接続 (GPIO)

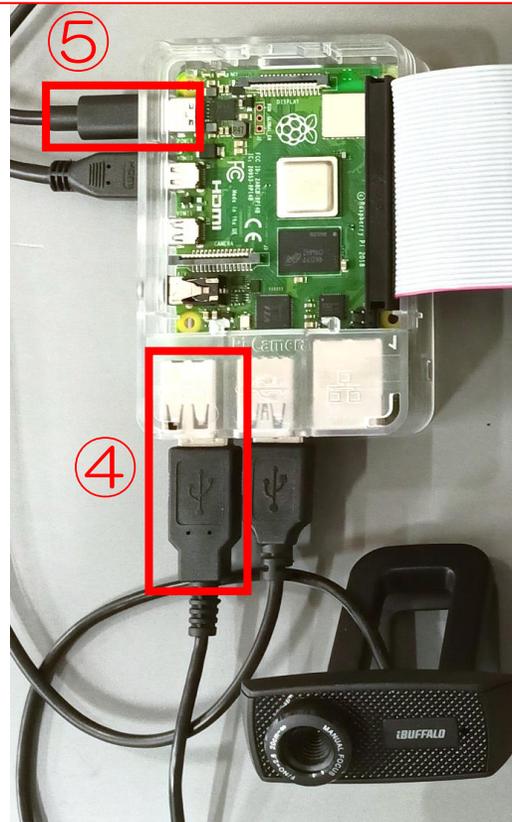


茶 : 16(#12) - 26  
紫 : 17(GND) - 24



センサ回路の作製完了 ⇒ 「IoT導入支援キット」 起動

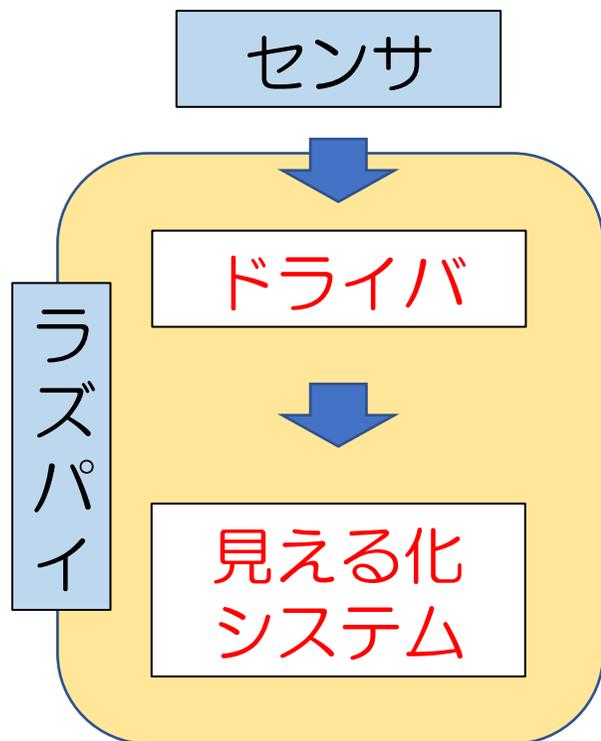
- ①HDMIケーブルを接続
- ②モニタ用ACアダプタを接続
- ③キーボード、マウスを接続
- ④カメラをラズパイに接続
- ⑤ラズパイ用ACアダプタを接続



# ③動作確認

ソフトウェア、センサ回路の動作確認

# 全体概要（ソフトウェア）



## ドライバ (Python)

- センサからデータを取得
- Node-RED用にデータ形式を変換

## 見える化システム (Node-RED)

- ドライバからデータを取得
- データの見える化（ブラウザ表示）

# 接続確認 (I2C センサ)



左から4番目のアイコンをクリックして  
LXTerminalを起動

```
pi@denshi-pi-001: ~ $ i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d
00:  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  29  --  --  --  --  --  --
30:  --  --  --  --  --  --  39  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  67  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --
```

コマンド『i2cdetect -y 1』を実行

距離センサのアドレス (29)

照度センサのアドレス (39)

温度センサのアドレス (67)

# 動作確認 (Pythonで書かれたプログラム実行)

コマンド

『node-red-stop』を実行する  
(ドライバーを使っているので止める)

```
pi@denshi-pi-027:~$ node-red-stop
Stop Node-RED
Use node-red-start to start Node-RED again
pi@denshi-pi-027:~$
```

コマンド

『python3 /home/pi/python/mcp9600.py』  
を実行 (温度センサ)

```
pi@denshi-pi-027:~$ python3 /home/pi/python/mcp9600.py
```

実行結果

**Ctrl** + **C** を押して止める

```
["ch0": 20.3125, "time": 1570088765]
["ch0": 20.375, "time": 1570088770]
["ch0": 20.4375, "time": 1570088775]
["ch0": 20.375, "time": 1570088780]
```

同様に

『python3 /home/pi/python/tof.py --sensing\_mode GOOD\_ACCURACY』 (距離センサ)

『python3 /home/pi/python/mpu9250.py』 (振動センサ)

『python3 /home/pi/python/TSL2561.py --gain 1 --integral 101』 (照度センサ)

を実行

# 動作確認 (見える化システム (Node-RED) を実行)

```
pi@denshi-pi-027:~$  
pi@denshi-pi-027:~$  
pi@denshi-pi-027:~$ node-red-start  
Start Node-RED
```

① コマンド  
『node-red-start』 を実行



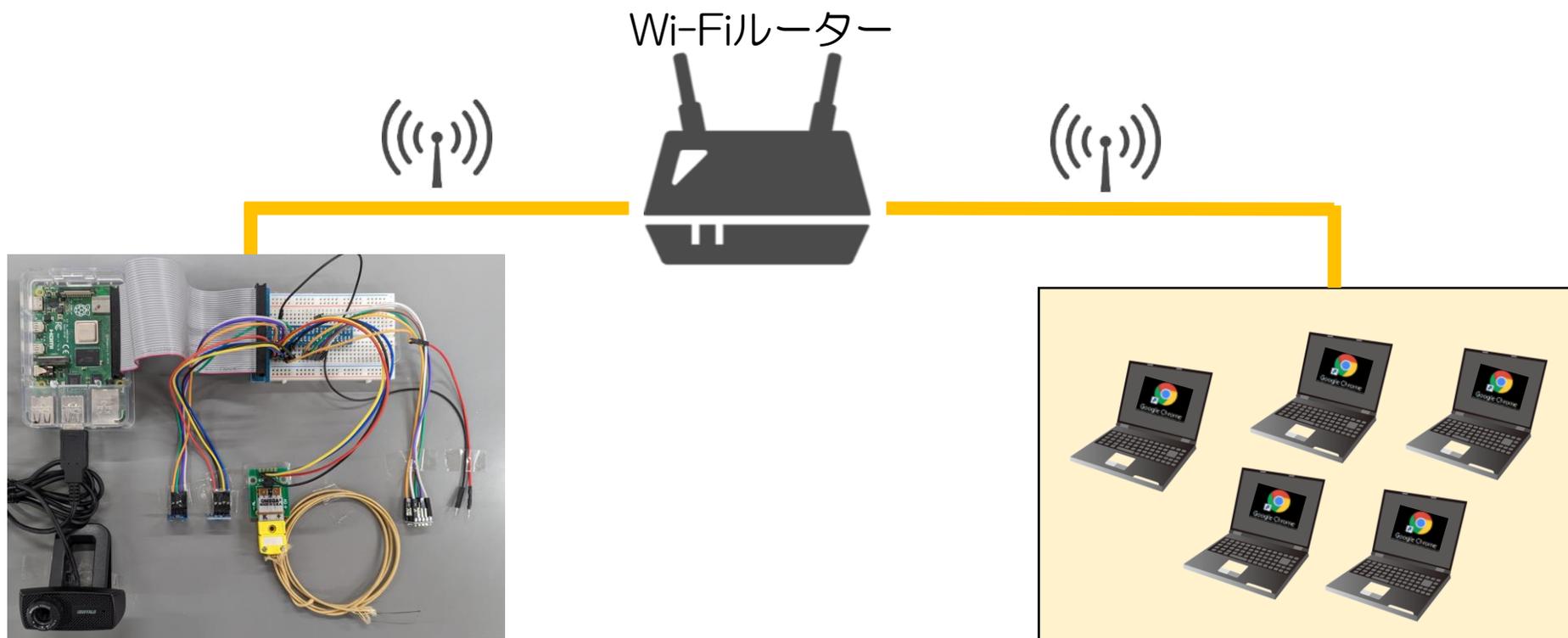
② 地球アイコンをクリックして起動

③ 『localhost:1880/ui』 を入力

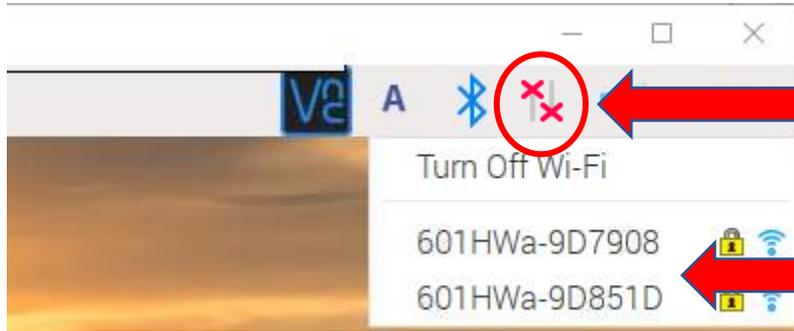
加速度	温度	距離	照度	GPIO(無電圧接点入力)
X軸 <b>0.5</b> m/s <sup>2</sup>	ch0 <b>22.4</b> °C	平均 <b>1,000</b> mm	可視 <b>613</b> CNT	GPIO12-GND <b>Open</b>
Y軸 <b>-8.0</b> m/s <sup>2</sup>	ch1 -- °C	最小 <b>1,000</b> mm	赤外 <b>180</b> CNT	スイッチ回数 <b>0</b> 回/sec
Z軸 <b>-5.6</b> m/s <sup>2</sup>	ch2 -- °C	最大 <b>1,000</b> mm	ゲイン: 1	モード <input type="checkbox"/> a接点
ch3 -- °C		モード: GOOD	積分時間: 402ms	
出力信号設定(加速度)	出力信号設定(温度)	出力信号設定(距離)	出力信号設定(照度)	出力信号設定(GPIO)
加速度軸 x軸	温度ch ch0	距離 平均	照度 可視光	
立上り[m/s <sup>2</sup> ] 0	立上り[°C] 24	立上り[mm] 60	立上り[CNT] 100	

# ④ 遠隔操作環境の構築

## 遠隔操作のためのWi-Fi接続環境を構築



# Wi-Fiへの接続（ラズパイ）

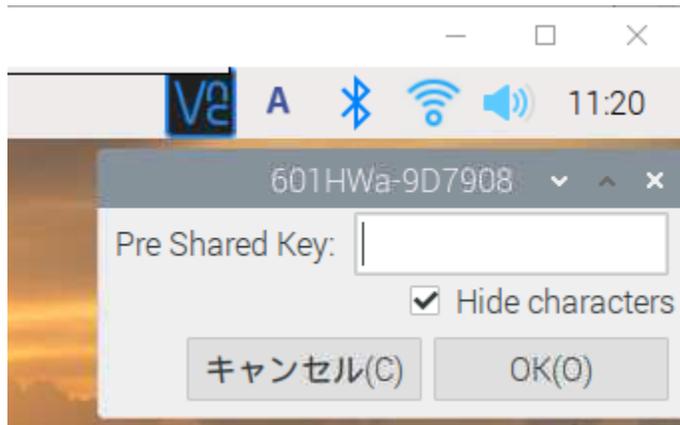


① クリックし、Turn On Wi-Fiを選択  
（既にONの場合は不要）

② 同じアイコンをクリックして少し待つ



③ 『SSID』を選んでクリック

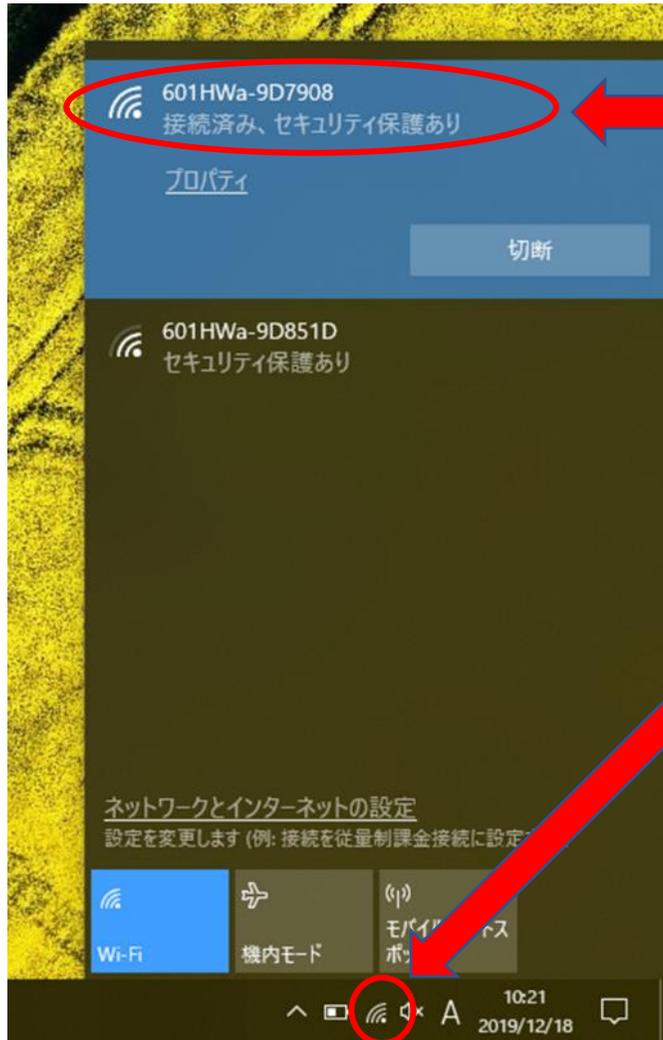


④ 『Password』を入力

⑤ OKをクリック

- ⑥ ①でクリックしたマークにカーソルを合わせて待つ（クリックしない）
- ⑦ Wlan Configured ●●●●/24と表示される ⇒ IPアドレスをメモ
- ⑧ 以後、ラズパイにつないでいるモニター、キーボード、マウスは使わないので外す

# Wi-Fiへの接続 (ノートPC)



② 『SSID』を選んでクリック

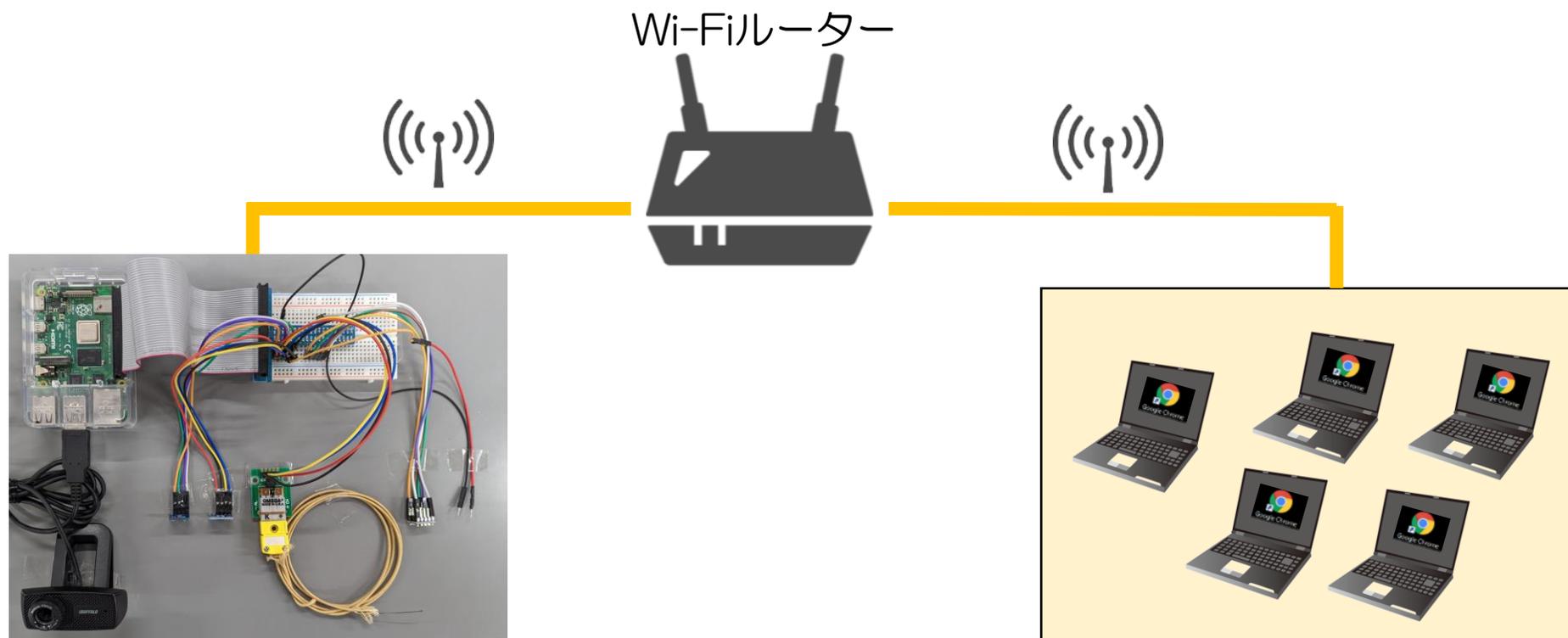
③ 『Password』を入力



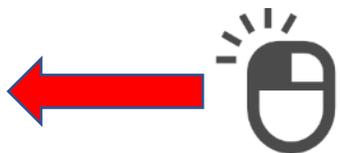
①地球マークをクリック

Wi-Fi接続完了  
(ラズパイ⇄ノートPC)

# PCのブラウザで『遠隔操作・監視』が可能



# ブラウザで表示 (ノートPC)



① アイコンをダブルクリックして起動



192.168.2.10x:1880/ui



② 『IPアドレス:1880/ui』 を入力

≡ ダッシュボード

加速度	温度	距離	照度	GPIO(無電圧接点入力)
X軸 <b>0.5</b> m/s <sup>2</sup>	ch0 <b>22.4</b> °C	平均 <b>1,000</b> mm	可視 <b>613</b> CNT	GPIO12-GND <b>Open</b>
Y軸 <b>-8.0</b> m/s <sup>2</sup>	ch1 -- °C	最小 <b>1,000</b> mm	赤外 <b>180</b> CNT	スイッチ回数 <b>0</b> 回/sec
Z軸 <b>-5.6</b> m/s <sup>2</sup>	ch2 -- °C	最大 <b>1,000</b> mm	ゲイン: 1	モード <input type="checkbox"/> a接点
	ch3 -- °C	モード: GOOD	積分時間: 402ms	
出力信号設定(加速度)		出力信号設定(温度)		出力信号設定(GPIO)
加速度軸 x軸	温度ch ch0	距離 平均	照度 可視光	
立上り[m/s <sup>2</sup> ] 0	立上り[°C] 24	立上り[mm] 60	立上り[CNT] 100	
立下り[m/s <sup>2</sup> ] 0	立下り[°C] 24	立下り[mm] 60	立下り[CNT] 100	
信号反転 <input type="checkbox"/>	信号反転 <input type="checkbox"/>	信号反転 <input type="checkbox"/>	信号反転 <input type="checkbox"/>	
GPIO17 カウンタ <b>ON</b> 3	GPIO06 カウンタ <b>OFF</b> 0	GPIO05 カウンタ <b>ON</b> 13	GPIO23 カウンタ <b>ON</b> 1	GPIO16 カウンタ <b>OFF</b> 10

カメラ 振動スペクトログラム

# ソフトウェア(ドライバ、見える化システム)のライセンスについて

適用ライセンス：Apache License, Version 2.0

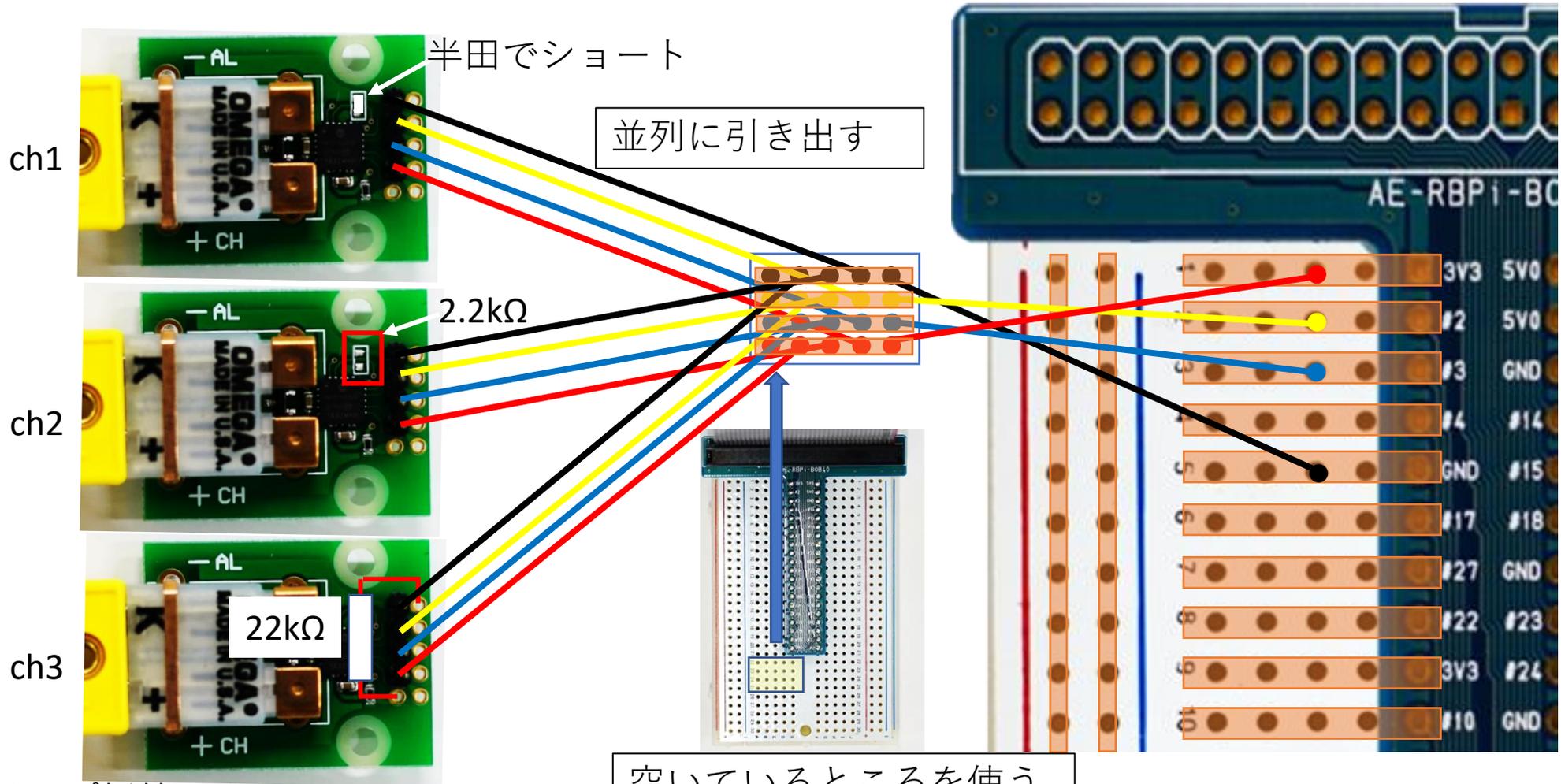
URL:<http://www.apache.org/licenses/LICENSE-2.0>

- 商用(私用)利用と修正、再配布が可能
- 修正、再配布の際は、著作権の表示、変更箇所の明示が必要
- 使用や配布に伴うトラブルについては、自己責任

# 用語説明

- ① ラズパイ：  
Raspberry Pi（ラズベリーパイ）は、ARMプロセッサを搭載したシングルボードコンピュータで、イギリスのラズベリーパイ財団によって開発されている。  
IoT導入支援キットのメインコンピュータで、正式名称は『RaspberryPi』である。
- ② Chrome：  
google製のウェブブラウザである。  
[https://www.google.com/intl/ja\\_jp/chrome/](https://www.google.com/intl/ja_jp/chrome/)
- ③ balena Etcher  
イメージファイルからSDカードにシステムを書き込むツールである。  
<https://www.balena.io/etcher/>
- ④ Node-RED：  
IBMによってIoTのために開発された、フローベースプログラミングツールである。  
IoT導入支援キットの開発言語として使用している。
- ⑤ I2C：フィリップス社が提唱した周辺デバイスとのシリアル通信の方式。電源・GND・SDA・SCLの4本の信号線で通信するものである。
- ⑥ SPI：同期式シリアル通信の一つ。電源・GND・MISO・MOSI・SCLK・CSの6本の信号線で通信するものである。
- ⑦ Python：汎用のプログラミング言語である。コードがシンプルで扱いやすく設計されており、C言語などに比べて、さまざまなプログラムを分かりやすく、少ないコード行数で書けるといった特徴がある。IoT導入支援キットのセンサードライバーに使用している。

# 温度センサの追加について



チップ抵抗、もしくは  
5番・10番端子に抵抗を繋ぐことで、  
チャンネルを指定